# PROFESSIONAL TRAINING REPORT

**at**

**Sathyabama Institute of Science and Technology**

**(Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

By

## HARIHARAN B P

**REG. NO. 39110373**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

**CHENNAI – 600119, TAMILNADU**

**NOVEMBER 2021**

|  |  |  |
| --- | --- | --- |
|  | **SATHYABAMA**  **INSTITUTE OF SCIENCE AND TECHNOLOGY** (DEEMED TO BE UNIVERSITY) **Accredited with Grade “A” by NAAC**  (Established under Section 3 of UGC Act, 1956)  JEPPIAAR NAGAR, RAJIV GANDHI SALAI  CHENNAI– 600119  [**www.sathyabama.ac.in**](http://www.sathyabama.ac.in) |  |

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **HARIHARAN B P (Reg. No: 39110373)** who carried out the project entitled “**SONG RECOMMENDER USING FACIAL EXPRESSION**” under my supervision from June 2021 to November 2021.

**Internal Guide**

## Mr. Babu,

**Head of the Department**



## Submitted for Viva voce Examination held on

**Internal Examiner External Examiner**

**DECLARATION**

I, **HARIHARAN B P** hereby declare that the project report entitled “**SONG RECOMMENDER USING FACIAL EXPRESSION”** done by me under the guidance of Mr. Babu is submitted with partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

**DATE:**

**PLACE: SIGNATURE OF THE CANDIDATE**

**ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D.**, **Dean**, School of Computing **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Mr. Babu for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

# ABSTRACT

“SONG RECOMMEDER SYSTEM USING EMOTION DETECTION” is an Android Application where a user will be able to play songs according to their mood(emotion). This application lets the user to select the language and using emotional recognition. “SONG” a well-known and very meaningful word, which is the one-stop you will go to calm your mind and all the surroundings will be gone when u dive into this virtual world full of your favorite songs where you listen to them and at-times you sing with the singer to enjoy the song to the fullest.

Music plays an important role in an individual’s life. It is an important source of entertainment and is often associated with a therapeutic role. With the Advent of technology and continuous Advancements in multimedia, sophisticated music players have been designed and have been enriched with numerous features. The requirements of an individual, a user sporadically suffered through the need and desire of browsing through his playlist, according to his Mood and emotions. ‘This task was labor intensive and an individual often faced the dilemma of landing. At appropriate list of songs. Hence, this application. Can be used which dynamically suggest songs from. Your playlist according to the mood the user is feeling for his/her satisfaction.

As we all know, human emotion played a vital role in recent times and will play in coming future. Emotion is based on human feelings which can be both expression (most times) or words. It expresses the individual’s behavior which can be in different forms.

But there isn’t a lot of applications which combines both these concepts. So, we have come-up with an idea to this develop this application. The objective of the project is to extract feature from human face and detect emotion and play music according to the emotion detected. Facial expressions are captured locally using a camera device. Then the facial structures are compared and the songs are played based on what the emotion is.

**TABLE OF CONTENTS**

**CHAPTER No. TITLE PAGE No.**

ABSTRACT i

LIST OF FIGURES ii

LIST OF TABLES iii

LIST OF ABBREVIATIONS iv

1. **INTRODUCTION 9**
   1. FACIAL RECOGNITION 9
   2. WHY THIS APPLICATION? 9
   3. COMPUTER VISION USAGE 9
2. **AIM AND SCOPE OF THE PRESENT 10**

**INVESTIGATION**

* 1. AIM 10
  2. SCOPE 11
  3. INVESTIGATION 11

1. **EXPERIMENTAL OR MATERIALS**

**AND METHODS, ALGORITHMS USED 12**

3.1 EXPERIMENTAL OR MATERIALS 12

3.2 METHODS 14

3.3 ALGORITHMS 21

1. **RESULTS AND DISCUSSION, PERFORMANCE 24**

**ANALYSIS**

* 1. RESULTS AND PERFORMANCE

ANALYSIS OF DATASET 24

4.2 DISCUSSIONS 26

1. **SUMMARY AND CONCLUSIONS 28**

**REFERENCES 29**

**APPENDIX 30**

1. SCREENSHOTS 30
2. SOURCE CODE 31

# LIST OF FIGURES

**FIGURE No. FIGURE NAME PAGE No.**

1.1 Project Outline Diagram for the application 10

3.1 Sequence Diagram for Song Recommender 12

3.2 Flow Chart for App process 16

3.3 Ideation Map of Song Recommender 17

3.4 Over fitting graph for datasets 21

3.5 RMS-Prop Expression 24

3.6 Adam Expression 24

4.1 Model Accuracy 26

4.2 Model Loss 26

# LIST OF ABBREVIATIONS

ABBREVIATION EXPANSION

USD United States Dollar

CV Computer Vision

OpenCV Open-Source Computer Vision Library

VSCode Visual Studio Code

**CHAPTER 1**

**INTRODUCTION**

1.1 FACIAL RECOGNITION

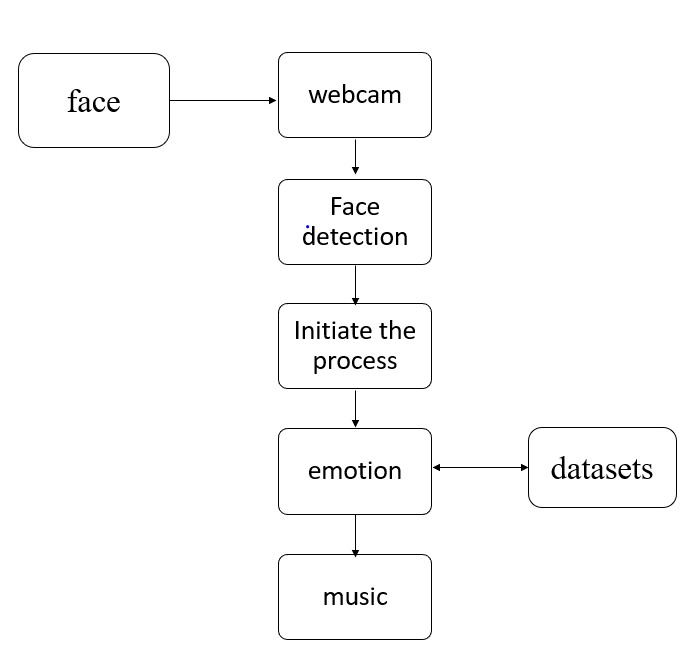
Face detection and recognition has been an on-going research area for the last 50 years, with concluding results being obtained starting with the late 90s. The fast development of facial recognition technology allowed it to be used in a variety of areas like assisted living, health monitoring, access control, authentication, ID/passport control and fraud prevention, security/law enforcement, surveillance systems, attendance tracking and many others. According to a report published by Markets and Markets in 2017, the global facial recognition market was estimated at 3.37 billion USD in 2016 and it is expected to grow up to 7.76 billion USD by 2022, with an annual growth rate of 13.9%.

1.2 WHY THIS APPLICATION?

In this concept music is recommended to the user by detecting the real time capturing of user’s emotions. Existing technique were using collaboration technique which will use previous user data to recommend music and This technique requires lot of manual work so, we proposed a system (as shown in fig 1.1) to arrange different music in different categories such as happy, sad or angry etc.

1.3 COMPUTER VISION USAGE.

Computer Vision (CV) is the field of study that helps computers to study using different techniques and methods so that it can capture what exists in an image or a video. There are a large number of applications of computer vision that are present today like facial recognition, driver less cars, medical diagnostics, etc. We will discuss one of the interesting applications of CV that is Emotion Detection through facial expressions. CV can recognize and tell you what your emotion is by just looking at your facial expressions. It can detect whether you are angry, happy, sad, etc.



* 1. Project Outline Diagram for the application

**CHAPTER 2**

**AIM AND SCOPE OF THE PRESENT INVESTIGATION**

2.1 AIM:

Emotion-Based-music-player is a music player integrated with an application which has the capability to detect emotions I.e., the face of user with the help of machine learning algorithm using python. Based on the detected user’s mood song list will be displayed/recommend to the user.

In this application image of a person is captured using a real time machine that has the access to the local machinery. And depending on the captured image it compares the data sets stored in the database which are already saved in the local device, through processing it defines the present mood of the user in numerical form based on this music will be played. Other than that, we have some common features that are queue playlist so that we can have an individual playlist and the last one is random. For this we have used libraries like Open CV, keras, tensor flow etc.

2.2 SCOPE

This system is mainly proposed because music play a vital role in recent times that is to reduce stress. so, in order to detect the emotion, we are using face as a main source of data because normally face expression defines the Emotion so according to the mood, we play the music that it can change the user’s mood.

In this project, we presented a model to recommend a music-based om the emotion based detected from the facial expression. This project proposed designed & developed an emotion-based music recommendation system using face recognition System one that has the power to heal any stress or any kind of emotions.

Recent development promises a wide scope in developing an emotion-based music recommendation system. Thus, the proposed system presents Face based emotion recognition system to detect the emotions and play music from the emotion detected.

2.3 INVESTIGATION

There are a lot of music/song recommender system but, none has the ability to give songs based on the present human emotion. So, we came up with the idea to use these resources to develop an app which uses Python modules and song APIs to recommend songs to the user based on his current emotion.

Yes, there are apps which takes user’s manual input and recommends song but, this system reduces the user’s work and makes his search low. So, it struck us that we can give it a go and make it a reality for the users as an instant song listening app.

**CHAPTER 3**

**EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED**

3.1 EXPERIMENTAL OR MATERIALS

Materials Used:

1. VSCode:

Visual Studio Code is an Integrated Development Environment (IDE) made by Microsoft for Windows, Linux, and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git

Extensions used:

1. Flutter VSCode extension: It is an Extension used for running flutter-based projects in VSCode.
2. Python extension

HARDWARE REQUIREMENTS:

1) ANDRIOD DEVICE/EMULATOR

2) Android 8 or more.

3) Min of 4GB RAM in device.

Tools used/software requirements:

1. OpenCV:

OpenCV is the library we will be using for image transformation functions such as converting the image to grayscale. It is an open-source library and can be used for many image functions and has a wide variety of algorithm implementations. C++ and Python are the languages supported by OpenCV. It is a complete package which can be used with other libraries to form a pipeline for any image extraction or detection framework. The range of functions it supports is enormous, and it also includes algorithms to extract feature descriptors.

1. Python:

Python is a powerful scripting language and is very useful for solving statistical problems involving machine learning algorithms. It has various utility functions which help in pre-processing. Processing is fast and it is supported on almost all platforms. Integration with C++ and other image libraries is very easy, and it has in-built functions and libraries to store and manipulate data of all types. It provides the pandas and NumPy framework which helps in manipulation of data as per our need. A good feature set can be created using the NumPy arrays which can have n-dimensional data.

1. Flutter:

It is framework developed by google which uses dart programming language to develop android, IOS applications.

1. TensorFlow:

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

1. Dart:

Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.

1. Keras module:

Keras is a neural network library. Keras provides only high-level APIs. Keras is built in Python which makes it way more user-friendly

3.2 METHODS

1. Facial Detection — Ability to detect the location of face in any input image or frame. The output is the bounding box coordinates of the detected faces.

2. Facial Recognition — Compare multiple faces together to identify which faces belong to the same person. This is done by comparing face embedding vectors

3. Emotion Detection — Classifying the emotion on the face as happy, angry, sad, neutral and surprise.

3.2.1 SYSTEM ARCHITECTURE:

In this project, by running the main it will trigger an XML file that then OpenCV helps in capturing images from the webcam as well as for processing purposes. We have made the implementation of the CNN along with OpenCV and TensorFlow for classification. And mobileNet using ImageNet to train the model and store it in a model-file (XML). While using a player it uses for prediction for the emotion which will show you the main media player in the app. In this, the app contains 2 options one for emotion-based detection and the other for random selection of song. The 3 main processes are capturing, detection and playing of the music. This system, describes the facial expressions using detection and combination of spatial expressions. After Feature Extraction, the Emotions are classified into different forms I.e., Happy, Angry, Sad and neutral face, etc. The emotions that are transferred to last step are in numerical form and the music is played from the emotions that are detected. The main objective of face detection technique is to identify the face. The sound module is used to access the local sound-playing machinery that is used. Fig 3.1 shows the overall sequence of execution of the app. MODULE DESCRIPTION

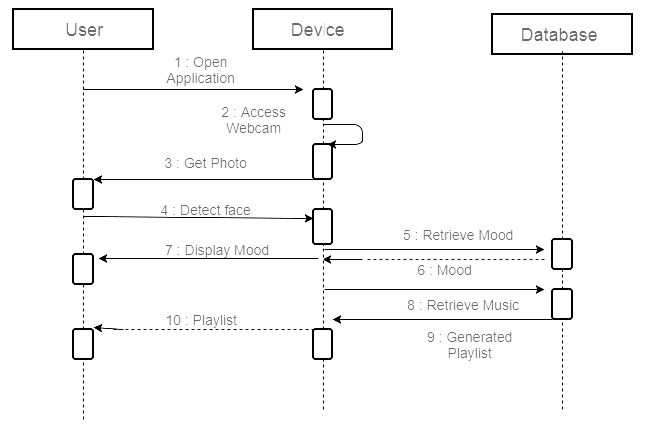
The proposed system majorly involves modules: -

A. Face extraction module

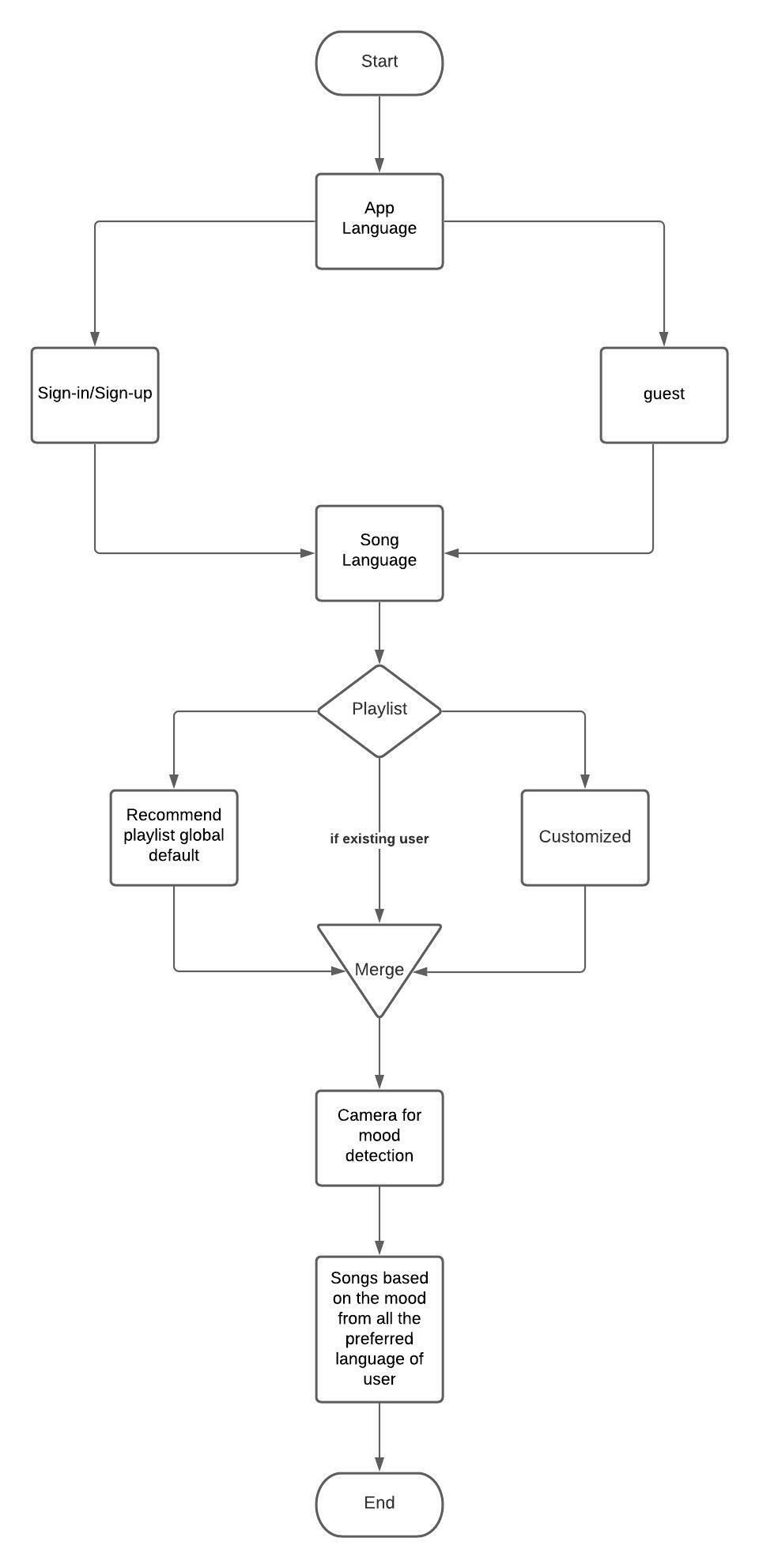
B. Emotion recognition module

C. Audio extraction module

D. Emotion-audio recognition module



3.1 Sequence Diagram for the Song application



3.2 Flow Chart for App process

3.2.2 PROCESS

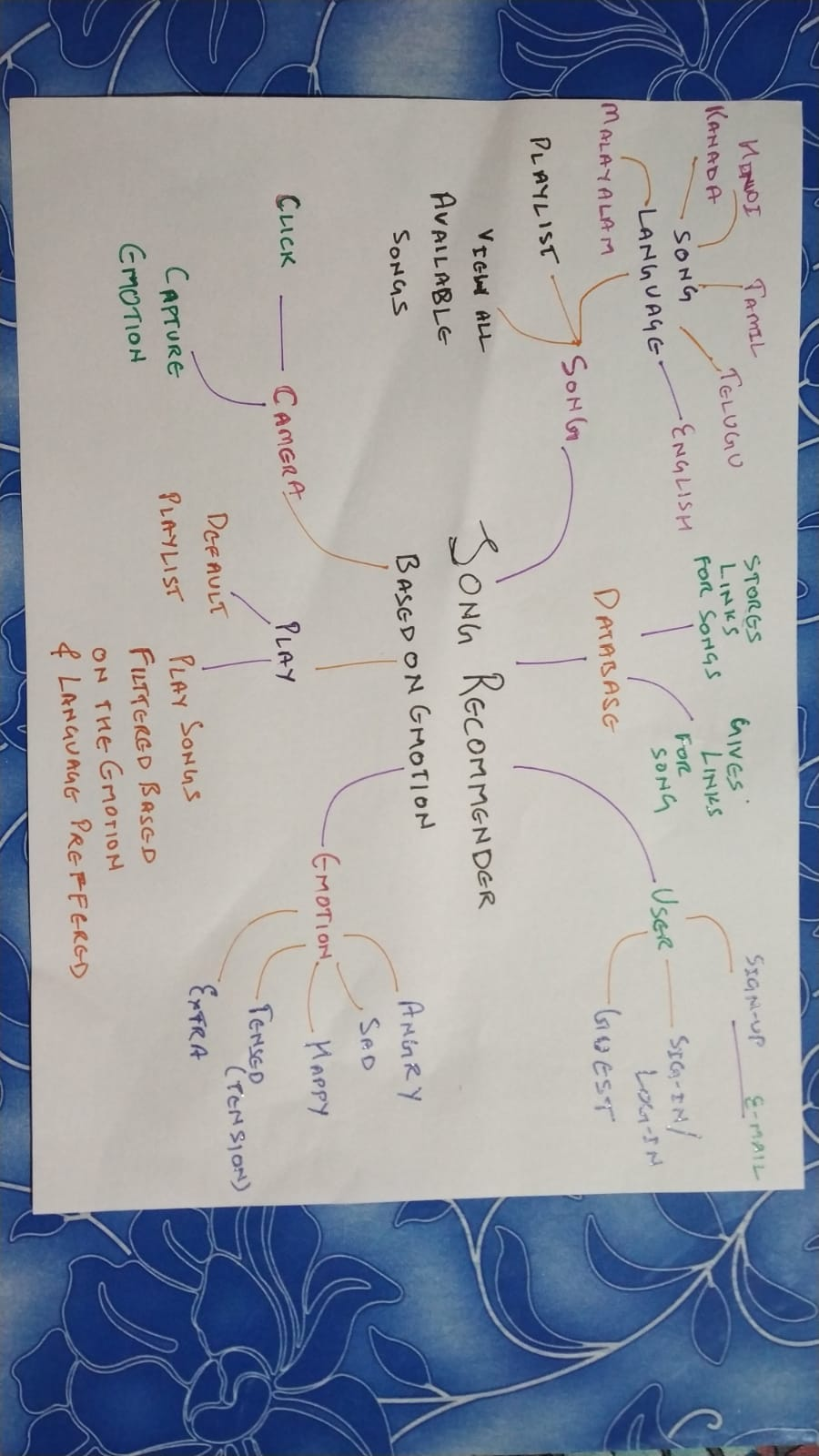
1. FACE CAPTURING:

The fig 3.2 shows the workflow of the application. Main objective of this session is to capture images so here we are using the common device i.e., webcam or can use any other physical devices. For that purpose, we are using the computer vision library. This makes it easier to integrate it with other libraries. In the initial process when the execution starts, it starts to access the camera and captures the image for further process and emotion detection. We authenticate and classify the images and we use positive images that actually contain images in order to train the classifier. The classified images are taken as a part of the model.

1. FACE DETECTION:

The face recognition is considered as one of the best ways to determine a person’s mood. This image processing system is used for reducing the face space dimensions and to obtain the feature of the image characteristics, we use this because it maximizes the training process in between classes.

By comparing the data sets that each emotion is compared to stored images and scale gives the exact emotion so that it can play the music based on the recommendation made by the system.



3.3 Ideation Map of Song Recommender

This algorithm helps to process for image recognition to classify the expression that implies the emotion of the user. Along with OpenCV mainly its emphasis on the class specific transformation matric, so they don’t take illustrative images as the subject. Emotion is mainly concluded by the model that the value evaluated from the process can help us to deduce the mood of the user.

1. EMOTION CLASSIFICATION:

When the face is detected successfully, a box will appear as and it overlay the camera image input to extract the face and for the further analysis in the next step. The images that are extracted previously will processed using the function. The code will extract the facial spatial positions from the face image and it is based on the pixel’s intensity values that are indexed at each point. As described in fig 3.3.

It performs the comparison between the input data and with stored one so it can predict the class that contain the emotion. If it contains one of the emotions. The songs which represent the particular emotion will be selected.

1. MUSIC RECOMMENDATION:

The input images that is acquired is from the camera and is used to capture real-time images. It compares the values that are present as a threshold in the code. The values will be transferred to perform the service. The songs will be played from the detected emotion. The emotions are assigned to every song. All the songs and the emotions are numbered and arranged. The mechanism. Obtained are being compared to the values that are present as a threshold. We can make a playlist as the other usual music software’s and the last one is the random mode it is for random picking of songs.

3.2.3 TRANSFER LEARNING

Transfer learning generally refers to a process where a model trained on one problem is used in some way on a second related problem. In deep learning, transfer learning is a technique whereby a neural network model is first trained on a problem similar to the problem that is being solved. One or more layers from the trained model are then used in a new model trained on the problem of interest. Transfer learning has the benefit of decreasing the training time for a neural network model and can result in lower generalization error.

Transfer learning involves using models trained on one problem as a starting point on a related problem. Transfer learning is flexible, allowing the use of pre-trained models directly, as feature extraction preprocessing, and integrated into entirely new models. Keras provides convenient access to many top performing models on the ImageNet image recognition tasks such as VGG, Inception, and ResNet.

3.2.4 DATA AUGMENTATION:

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. Training a neural network model on more data can result in more skillful models, and the augmentation techniques can create variations of the images that can improve the ability of the fit models to generalize what they have learned from new images.

Modern deep learning algorithms, such as the convolution neural network, or CNN, can learn features that are invariant to their location in the image. Nevertheless, augmentation can further aid in this transform invariant approach to learning and can aid the model in learning features that are also invariant to transforms such as left-to-right to top-to-bottom ordering, light levels in photographs, and more.

Image data augmentation is applied only to the training dataset, and not to the validation or test dataset. This is different from data preparation such as image resizing and pixel scaling; they must be performed consistently across all datasets that interact with the model.

3.2.5 BATCH NORMALIZATION:

Batch normalization normalizes a layer input by subtracting the mini-batch mean and dividing it by the mini-batch standard deviation. Mini-batch refers to one batch of data supplied for any given epoch, a subset of the whole training data. The normalization ensures that the inputs have a mean of 0 and a standard deviation of 1, meaning that the input distribution to every neuron will be the same, thereby fixing the problem of internal co variate shift and providing regularization.

If each layer is normalized, the weight changes made by the previous layer and noise between data is partially lost, as some non-linear relationships are lost during normalization. This can lead to sub optimal weights being passed on. To fix this, batch normalization adds two trainable parameters, which can scale and shift the normalized value.

3.2.6 OPTIMIZATION:

Optimizer are algorithms or methods used to minimize an error function (loss function) or to maximize the efficiency of production. Optimizer are mathematical functions which are dependent on model’s learnable parameters i.e., Weights & Biases. Optimizer help to know how to change weights and learning rate of neural network to reduce the losses.

1. Early Stopping:

A major challenge in training neural networks is how long to train them.

Too little training will mean that the model will underfit the train and the test sets. Too much training will mean that the model will overfit the training dataset and have poor performance on the test set. We train on the training datasets but have to stop training at the point when performance on a validation dataset starts to degrade. This simple, effective, and widely used approach to training neural networks is called early stopping.

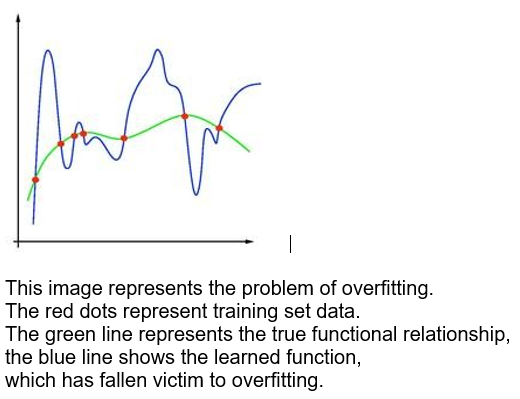
There are three elements to using early stopping; they are:

• Monitoring model performance.

• Trigger to stop training.

• The choice of model to use

This overfitting (fig 3.4) of the training datasets will result in an increase in generalization error, making the model less useful at making predictions on new data.



3.4 Over fitting graph for datasets

1. Loss Function:

The purpose of loss functions is to compute the quantity that a model should seek to minimize during training.

* 1. Entropy:

Entropy is the number of bits required to transmit a randomly selected event from a probability distribution.

* 1. Loss Categorical Cross Entropy:

Categorical cross entropy is a loss function that is used in multi-class classification tasks. It is designed to quantify the difference between two probability distributions.

SoftMax is the only activation function recommended to use with the categorical cross entropy loss function.

3.2.7 TRAINING AND TESTING ERROR:

Training error is the error that you get when you run the trained model back on the training data. Remember that this data has already been used to train the model and this necessarily doesn't mean that the model once trained will accurately perform when applied back on the same training data.

Test error is the error you get when you run the trained model on a set of data that it has previously never been exposed to. This data is often used to measure the accuracy of the model before it is shipped to production.

3.3 ALGORITHMS

1. First, the haar-cascade method is used to detect faces in each frame of the webcam feed.
2. The region of image containing the face is resized to 224x224 and is passed as input to the CNN.
3. The network outputs a list of SoftMax scores for the seven classes of emotions.
4. The emotion with maximum score is displayed on the screen.

Algorithms used:

1. CNN (Convolution Neural Network):

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of artificial neural network, most commonly applied to analyze visual imagery.

1. HaarCascade:

It is a machine learning algorithm to categorize objects in a captured image. It is mainly used for object detection. The cascade classifier has different stages of collection which resembles from weak learners. These weak classifiers are the simplest form classifiers that have a name called boosting. If the label ranges in positive state, then it goes to the next stage showing the result. These have a positive side and a negative side where they identify the images according to the labels. These have a set of positive images over negative images on various stages. As images with higher resolution has greater quantity are preferred as better-quality results.

1. MobileNet:
   1. Image Net:

ImageNet is an image database organized according to the WordNet hierarchy, in which each node of the hierarchy is depicted by hundreds and thousands of images.

* 1. MobileNet Architecture:

MobileNet is a lightweight architecture. MobileNet is a streamlined architecture that uses depth wise separable convolutions to construct lightweight deep convolutional neural networks and provides an efficient model for mobile and embedded vision applications The structure of MobileNet is based on depth wise separable filters.

1. convolutional layer with stride 2

2. depthwise layer

3. pointwise layer that doubles the number of channels

4. depthwise layer with stride 2

5. pointwise layer that doubles the number of channels

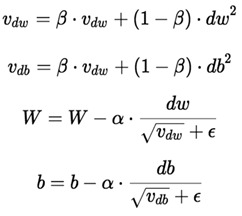
Depth wise separable convolution filters are composed of depth wise convolution filters and point convolution filters. The depth wise convolution filter performs a single convolution on each input channel, and the point convolution filter combines the output of depth wise

convolution linearly with 1 ∗ 1 convolutions.

1. OPTIMIZATION:
   1. RMS-Prop (Root Mean Square Propagation):

RMSprop is a gradient-based optimization technique used in training neural networks (fig 3.5) gradients of very complex functions like neural networks have a tendency to either vanish or explode as the data propagates through the function. Rmsprop was developed as a stochastic technique for mini-batch learning.

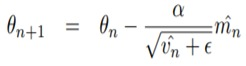
It deals with the issue by u moving average of the squared gradients to normalize the gradient. This normalization balances the step size (momentum), decreasing the step for large gradients to avoid exploding and increasing the step for small gradients to avoid vanishing.



3.5 RMS-Prop Expression

* 1. Adam (Adaptive Moment Estimation):

Adam optimizer is a first-order-gradient-based algorithm of stochastic objective functions (fig 3.6), based on adaptive estimates of lower-order moments. Adam is one of the latest state-of-the-art optimization algorithms being used by many practitioners of machine learning. The first moment normalized by the second moment gives the direction of the update. It is a method that computes adaptive learning rates for each parameter. It stores both the decaying average of the past gradients, similar to momentum and also the decaying average of the past squared gradients



3.6 Adam Expression

For adaptive methods like Adam and RMSprop, the learning rate is variable for each parameter. we have seen simple and computationally efficient algorithms for gradient-based optimization. the RMSprop and ADAM optimizers are straightforward and easy to implement.

1. LOSS CATEGORICAL CROSS ENTROPY:

Cross-entropy builds upon the idea of entropy from information theory and calculates the number of bits required to represent or transmit an average event from one distribution compared to another distribution

The cross-entropy between two probability distributions, such as Q from P, can be stated formally as:

H(P, Q)

Where,

H() is the cross-entropy function,

P may be the target distribution and

Q is the approximation of the target distribution.

Cross-entropy can be calculated using the probabilities of the events from P and Q, as follows:

H(P, Q) = – sum x in X P(x) \* log(Q(x))

Where,

P(x) is the probability of the event x in P,

Q(x) is the probability of event x in Q and

log is the base-2 logarithm, meaning that the results are in bits. If the base-e or natural logarithm is used instead, the result will have the units called Nats.

This calculation is for discrete probability distributions, although a similar calculation can be used for continuous probability distributions using the integral across the events instead of the sum.

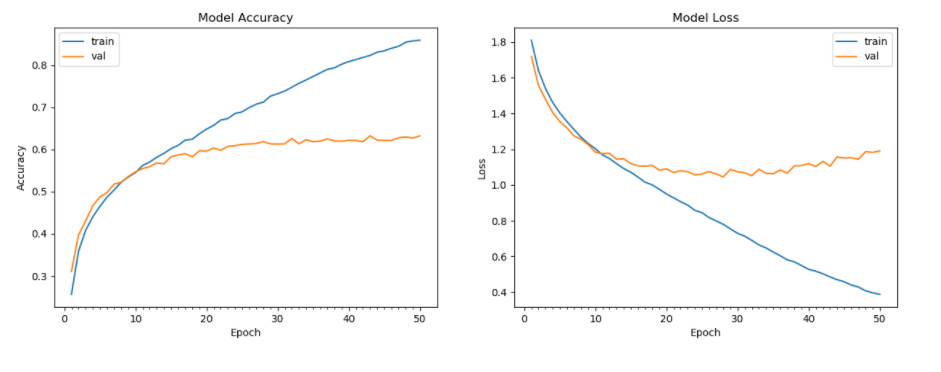
The result will be a positive number measured in bits and will be equal to the entropy of the distribution if the two probability distributions are identical.

**CHAPTER 4**

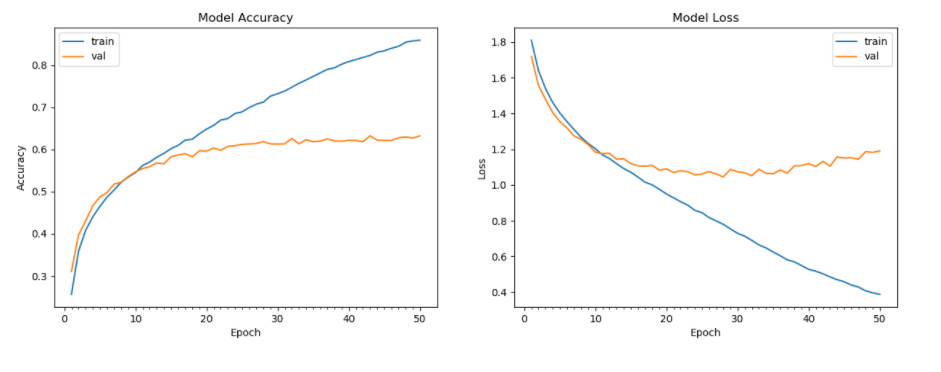
**RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS**

4.1 RESULTS AND PERFORMANCE ANALYSIS OF DATASET:

This implementation by default detects emotions on all faces in the webcam feed. With a simple 4-layer CNN, the test accuracy reached 63.2% in 50 epochs refer fig 4.1 and 4.2.



* 1. Model Accuracy



4.2 Model Loss

Given the high complexity of transfer learning models and relatively small size of our datasets, we also experienced overfitting while training. Although we added 30% dropout for the last three layers, our ResNet transfer learning model quickly over fit to the training set with train accuracy starting to flatten after only 25 epochs.

The model consists of three stages of convolutional and max-pooling layers, followed by an FC layer of size 1024 and a softmax output layer. The convolutional layers use 32, 32, and 64 filters of size 5x5, 4x4, and 5x5, respectively. The max-pooling layers use kernels of size 3x3 and stride 2.ReLU was utilized as the activation function. To improve performance, we also added batch norm at every layer and 30% dropout after the last FC layer. To fine tune the model, optimizing the cross-entropy loss with a momentum of 0.3. The initial learning rate, batch size, and weight decay are fixed at 0.2, 32, and 0.0001, respectively. The learning rate is halved if the validation accuracy does not improve.

Epoch is when an ENTIRE dataset is passed forward and backward through the neural network only ONCE. So, the ideal epochs for training a model is 25(minimum most cycle) here, we acquired the accuracy in 50 cycles. So, from the graph we can conclude that as the no. of cycles increases the accuracy increases and since the model is gaining knowledge as the cycle increases the model loss decreases.

4.2 DISCUSSIONS:

* The [original FER2013 dataset in Kaggle](https://www.kaggle.com/deadskull7/fer2013) is available as a single csv file. I had converted into a dataset of images in the PNG format for training/testing and provided this as the dataset in the previous section.
* FER2013 is a well-studied dataset and has been used in ICML competitions and several research papers. It is one of the more challenging datasets with human-level accuracy only at 65±5% and the highest performing published works achieving 75.2% test accuracy. Easily downloadable on Kaggle, the dataset’s 35,887 contained images are normalized to 224,224 pixels in grayscale. FER2013 is, however, not a balanced dataset
* Model — we built a 5 layered Convolutional Neural Network (CNN) in Keras and used Image augmentations, transfer learning, data augmentation, batch normalization and optimization algo to improve model performance.
* The original NJU\_MUSIC\_MOOD dataset from Kaggle consists of 100 songs under each category and the application will made in such a way that based on the emotion we got we will filter and recommend those songs based on user preferences in settings.

Step 1: The user gives input, which is in the form of the image captured using the camera.

Step 2: The image gets analysed by our model and gets classified as a happy, sad, neutral or angry emotion.

Step 3: The data gets extracted and detected with the training datasets, which are fer-13 datasets.

Step 4: The playlist or songs are chosen according to the facial mood recognition of the user.

Step 5: The music gets played to boost the user’s mood after successful detection of the sentiments

* Advantages of Proposed System

1. Users don’t want to select song manually.
2. No need of playlist.
3. Users don’t want to classify the songs based on the emotions

**DATASET IMAGES:**

**CHAPTER 5**

**SUMMARY AND CONCLUSIONS**

5.1 SUMMARY

The application aims to provide a simpler, additional hardware-free and reliable emotion-based music system to the operating system users. The Emotion-based music program would help people who are searching for music driven on the emotion and emotional behavior. It could help to reduce the search time for music and thus reduce the unnecessary computational time and thus increase the overall accuracy and efficiency of the system. The application solves the basic needs of music listeners without troubling them as existing applications do: it uses technology to increase the interaction of the system with the user in many ways. It eases the work of the end-user by capturing the image using a camera, determining their emotion, and suggesting a customized playlist.

5.2 CONCLUSIONS

This project has been developed to give us great advancement in the field of machine learning technology. Emotion Based Music player fulfils to sort out the music based on the emotions of the user such as whether it is happy or sad. So, totally our work aims to develop a player which is based on user need and it helps to revive in case of free time or leisure time if we want to hear music based on our current situation.

**REFERENCES:**

[1] Spotify — Song Prediction and Recommendation System  
 by : Sunku Sowmya Sree  
 Source: <https://medium.com/swlh/spotify-song-prediction-and-recommendation-system-b3bbc71398ad>

[2] Create Music Recommendation System Using Python  
 by : Ajinkya Khobragade  
 Source: <https://towardsdatascience.com/create-music-recommendation-system-using-python-ce5401317159>

[3] Facial Recognition using OpenCV

by: Atul Balaji

Source: <https://github.com/atulapra/Emotion-detection>

[4] A Complete Guide to Adam and RMSprop Optimizer

By: Sanghvirajit

Source: <https://medium.com/analytics-vidhya/a-complete-guide-to-adam-and-rmsprop-optimizer-75f4502d83be>

[5] Dataset source

By: Kaggle website

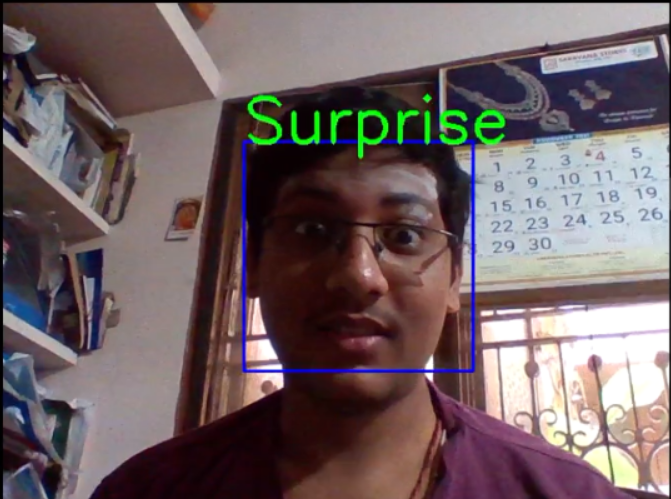
Source: <https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>

**APPENDIX**

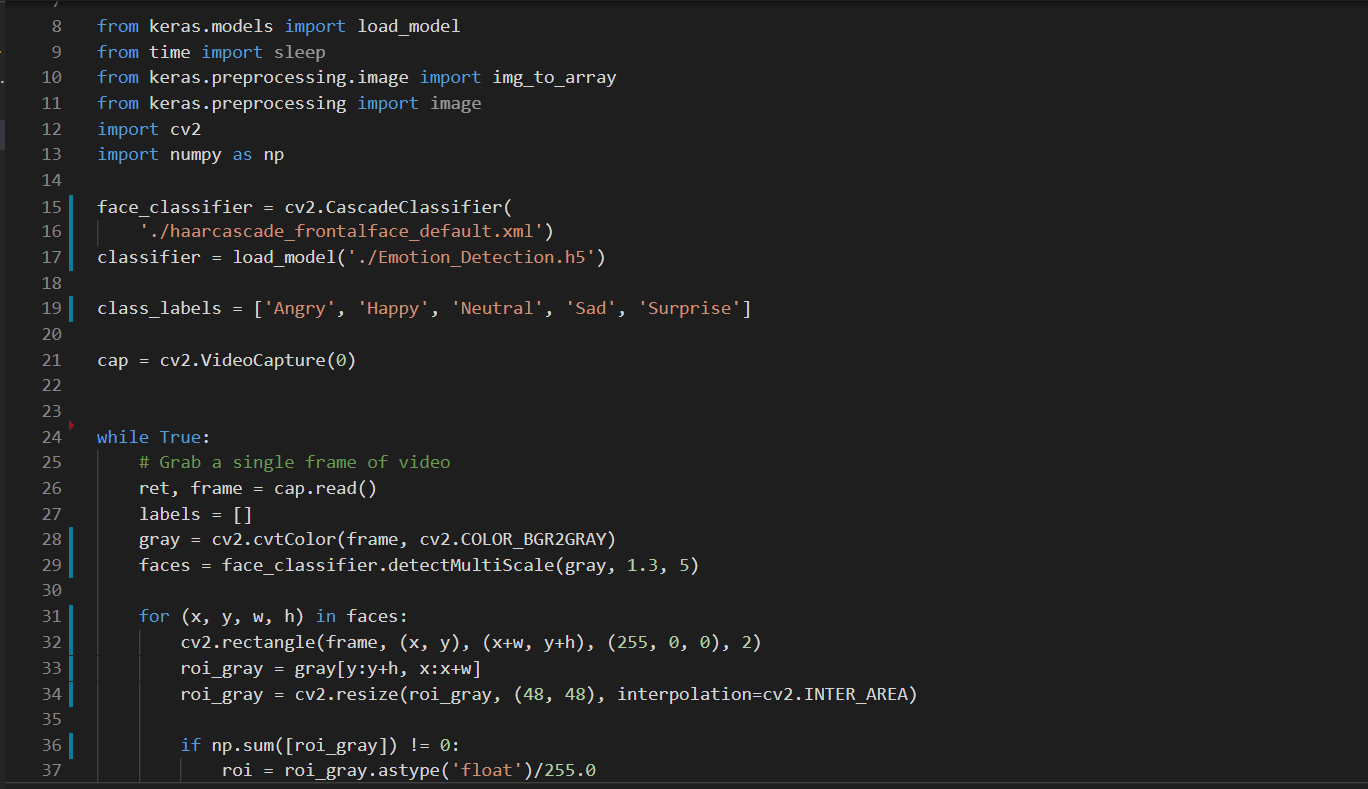
* 1. SCREENSHOTS

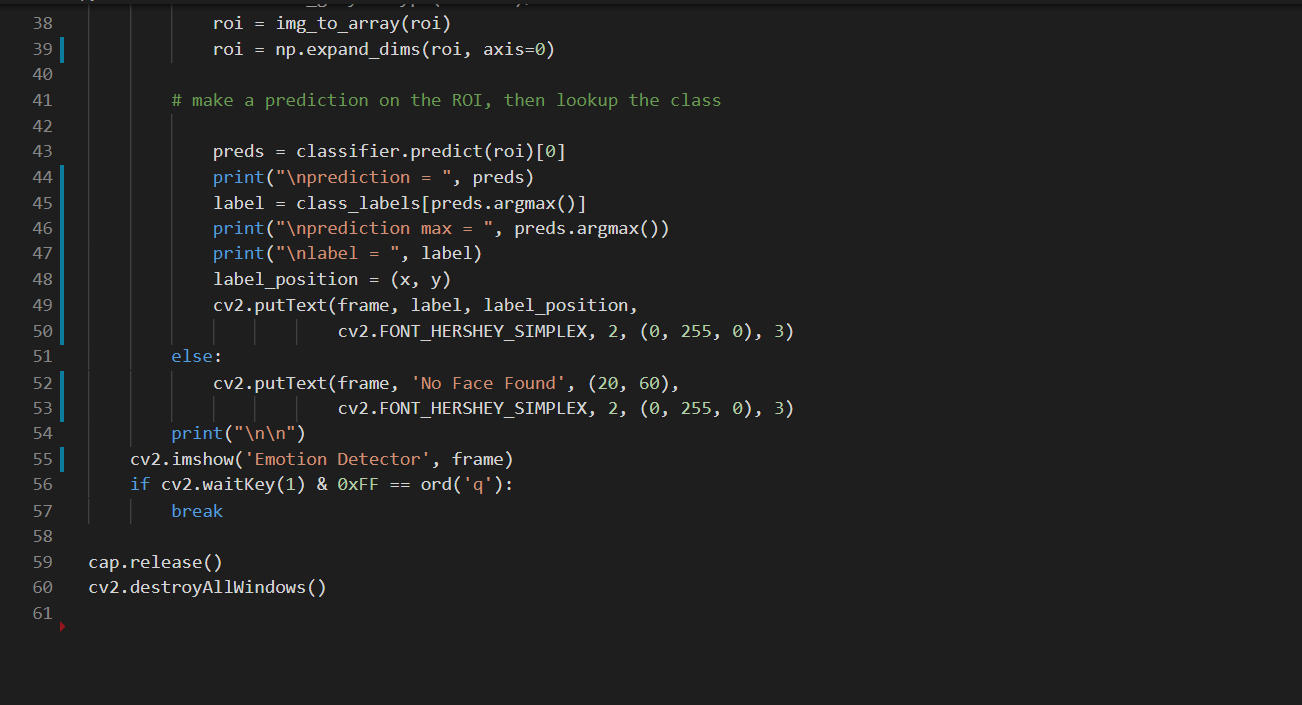




* 1. SOURCE CODE





from keras.models import load\_model

from time import sleep

from keras.preprocessing.image import img\_to\_array

from keras.preprocessing import image

import cv2

import numpy as np

face\_classifier = cv2.CascadeClassifier(

'./haarcascade\_frontalface\_default.xml')

classifier = load\_model('./Emotion\_Detection.h5')

class\_labels = ['Angry', 'Happy', 'Neutral', 'Sad', 'Surprise']

cap = cv2.VideoCapture(0)

while True:

# Grab a single frame of video

ret, frame = cap.read()

labels = []

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces = face\_classifier.detectMultiScale(gray, 1.3, 5)

for (x, y, w, h) in faces:

cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)

roi\_gray = gray[y:y+h, x:x+w]

roi\_gray = cv2.resize(roi\_gray, (48, 48), interpolation=cv2.INTER\_AREA)

if np.sum([roi\_gray]) != 0:

roi = roi\_gray.astype('float')/255.0

roi = img\_to\_array(roi)

roi = np.expand\_dims(roi, axis=0)

# make a prediction on the ROI, then lookup the class

preds = classifier.predict(roi)[0]

print("\nprediction = ", preds)

label = class\_labels[preds.argmax()]

print("\nprediction max = ", preds.argmax())

print("\nlabel = ", label)

label\_position = (x, y)

cv2.putText(frame, label, label\_position,

cv2.FONT\_HERSHEY\_SIMPLEX, 2, (0, 255, 0), 3)

else:

cv2.putText(frame, 'No Face Found', (20, 60),

cv2.FONT\_HERSHEY\_SIMPLEX, 2, (0, 255, 0), 3)

print("\n\n")

cv2.imshow('Emotion Detector', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()